Attorney Docket No. YO999-478

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): S.H. Maes et al.
Docket No.: YO999-478
Serial No.: 09/544,823
Filing Date: April 6, 2000
Group: 2142
Examiner: Douglas B. Blair

Title: Methods and Systems for Multi-Modal Browsing and Implementation of a Conversational Markup Language

---

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

Sir:

Applicants (hereinafter referred to as "Appellants") hereby appeal the current rejection of claims 1-91 of the above-referenced application.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corp., as evidenced by an assignment recorded June 16, 2000 in the U.S. Patent and Trademark Office at Reel 10854, Frame 0377. The assignee, International Business Machines Corp., is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals and interferences.

STATUS OF CLAIMS

Claims 1-91 are pending in the present application. Claims 1-12, 29, 36-56, 73, 80-87, 90 and 91 stand rejected under 35 U.S.C. §102(e). Claims 13-28, 30-35, 57-72, 74-79, 88 and 89 stand rejected under 35 U.S.C. §103(a). Claims 1-91 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a method of generating an application accessible by a user through one or more computer-based devices, comprising the steps of: representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser; and authoring the application using at least a portion of the interaction-based programming components.

Independent claim 44 recites apparatus for use in accessing an application in association with one or more computer-based devices, comprising: one or more processors operative to: (i) obtain the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and (ii) transcode the interaction-based programming components on a

component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modaility-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser.

Independent claim 90 recites a browser apparatus for use in providing access to an application by a user through one or more computer-based devices, comprising a machine readable medium containing computer executable code which when executed permits the implementation of the steps of: obtaining the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and transcoding the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modaility-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser.

Independent claim 91 recites an article of manufacture for use in generating an application accessible by a user through one or more computer-based devices, comprising a machine readable medium containing computer executable code which when executed permits the implementation of the steps of: representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent

of any modality and any modality-specific browser; and authoring the application using at least a portion of the interaction-based programming components.

Accordingly, principles of the invention defines a new application programming paradigm (Specification at page 6, line 26, through page 7, line 6). Existing application authoring approaches have adopted the concept of separating the content based aspects of an application from the presentation based aspects (Specification page 2, line 2, through page 6, line 2).

In accordance with the present invention, a new paradigm is introduced, illustratively embodied as a Conversational Markup Language (CML), which provides for separating application programming into content aspects, presentation aspects and interaction aspects. This new application programming paradigm is illustrated in FIG. 2 by reference numerals A, B and C.

By focusing on the interaction aspect of an application with respect to a user, an application may be written in a manner which is independent of the content/application logic and presentation. The "interaction-based programming components" recited in independent claims 1, 44, 90 and 91 provide such advantages since, as expressly recited, they are independent of content/application logic and presentation requirements associated with the application and they may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser. Examples of such presentation requirements and modality-specific renderings include visual-based (e.g., text and graphical) renderings, speech based renderings, and combinations thereof.

Thus, in accordance with the invention, a device operating with downloaded CML code can transcode to, for example, HTML and VoiceXML, substantially simultaneously so as to synchronize the multiple browsers providing the user with access to information. Such advantageous synchronization according to the invention is possible because the transcoding is done gesture by gesture with gesture identification. Thus, when an input/output event occurs in one modality, the browser knows what event occurred for what gesture and can immediately update all the supported modalities. This results in a very tight synchronization across modalities. Such synchronization is

also achieved due to the fact that the various modality-specific user interface dialogues, e.g., associated with a graphical user interface (GUI) browser or a speech browser, are generated from a single CML representation, on a gesture by gesture basis. Thus, the multiple user interfaces, e.g., GUI, speech, etc., are synchronized and continuously updated as a user interactively proceeds with one or the other modality (Specification, page 7, line 17, through page 8, line 14).

By way of example, FIG. 14 illustrates the operation of a multimodal browser according to an embodiment of the invention. As shown, an application developer writes an application, e.g., a light-weight application referred to as infoware, in CML. Infoware authored in CML is hosted by the a conversational shell (e.g., multimodal shell 62 of FIG. 13) that mediates amongst multiple modality specific browser components (e.g., visual browser 64 and speech browser 66 of FIG. 13). The multimodal shell may be thought of as a CML interpreter or processor. This is illustrated in FIG. 14 as block 70. User interaction proceeds by the CML interpreter mapping CML instances associated with the downloaded CML code to appropriate modality-specific languages such as HTML (block 77) and VoiceXML (block 78). These modality-specific representations render modality-specific versions of the dialog associated with the application. As illustrated in block 70, the nodes (A) and arrows (B) represent the declarative program in CML. The gestures in the CML program are represented by each of the nodes and the arrows represent the flow of the interaction/dialog with possible bifurcation points or loops. Each gesture is identified by a node ID (node_id) that allows appropriate identification of the activated gesture for synchronization between the different registered modalities. The node_id identifies the gesture so that the CML browser (i.e., the multimodal shell or virtual browser) knows where it is in the dialog flow and where to go from there (e.g., update the different modalities or send variables to the server and fetch a new CML page).

The transformation from CML to modality-specific representations 77 and 78 is governed by XSL transformation rules (or other transformation mechanisms, as mentioned above). These XSL rules are modality-specific. These transformations are handled by the presentation generation block 72 in accordance with the XSL rules 74 and the registration table 76. The registration table 76 is a repository of default gesture XSL transformation rules, as well as the specific rules that are extensions, application specific, device specific or user specific. In the process of mapping the CML instance to an appropriate modality-specific representation, the XSL rules add the necessary information needed to realize modality-specific user interaction. As an example, when translating

element select to VoiceXML, the relevant XSL transformation rule handles the generation of the grammar that covers the valid choices for that conversational gesture.

The process of transforming CML instances to modality-specific representations such as HTML may result in a single CML node mapping to a collection of nodes in the output representation. To help synchronize across these various representations, CML attribute node_id is applied to all output nodes resulting from a given CML node. When a given CML instance is mapped to different representations, e.g., HTML and VoiceXML by the appropriate modality-specific XSL rules, the shape of the tree in the output is likely to vary amongst the various modalities. However, attribute node_id allows us to synchronize amongst these representations by providing a conceptual backlink from each modality-specific representation to the originating CML node. This is graphically depicted in block 70 of FIG. 14.

As user interaction proceeds, variables defined in the environment by the current CML instance get bound to validated values. This binding happens first in one of the modality-specific representations (registered clients) 77 and 78. The modality-specific representation sends an appropriate message to the CML interpreter (multimodal shell) comprising of the updated environment and the node_id of the gesture that was just completed. Once the updated binding has been propagated to the CML interpreter, it messages all modality-specific representations with the node_id of the gesture just completed. Modality-specific representations update their presentation upon receiving this message by first querying the CML interpreter for the portion of the environment that affects their presentation.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

(1) Claims 1-12, 29, 36-56, 73, 80-87, 90 and 91 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,418,439 to Papierniak et al. (hereinafter "Papierniak").

(2) Claims 5 and 49 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of "New VXML Forum," posted at Cover Pages Hosted by Oasis (hereinafter "New VXML Forum").

(3) Claims 13-28, 34, 57-72 and 78 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,269,336 to Ladd et al. (hereinafter "Ladd").

(4) Claims 31, 32, 75 and 76 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,569,207 to Sundarsesan (hereinafter "Sundarsesan").

(5) Claims 30 and 74 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of World Wide Web Consortium document entitled "Extensible Stylesheet Language (XSL) version 1.0" (hereinafter referred to as "W3C XSL specification").

(6) Claims 33, 77, and 88 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,493,758 to McLain (hereinafter "McLain").

(7) Claims 35, 79, and 89 are rejected under 35 U.S.C. 103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,456,974 to Baker et al. (hereinafter "Baker").

ARGUMENT

(1) Claims 1-12, 29, 36-56, 73, 80-87, 90 and 91 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,418,439 to Papierniak et al. (hereinafter "Papierniak").

Regarding the §102(e) rejections of claims 1-12, 29, 36-56, 73, 80-87, 90 and 91, Appellants assert that Papierniak fails to teach or suggest all of the limitations in said claims for at least the reasons presented below.

It is well-established law that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Appellants assert that the rejection based on Papierniak does not meet this basic legal requirement, as will be explained below.

The present invention, for example, as recited in amended independent claim 1, recites a method of generating an application accessible by a user through one or more computer-based devices, comprising the following steps. Interactions that the user is permitted to have with the one or more computer-based devices used to access the application are represented by interaction-based

programming components. The interaction-based programming components are independent of content/application logic and presentation requirements associated with the application. Further, the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser. The application is then authored using at least a portion of the interaction-based programming components. Independent claim 91 recites similar limitations in an article of manufacture format. Independent claims 44 and 90 recite respective apparatus for accessing applications having similar limitations.

Accordingly, as explained above in the Summary section, a new paradigm is introduced, illustratively embodied as a Conversational Markup Language (CML), which provides for separating application programming into content aspects, presentation aspects and interaction aspects. By focusing on the interaction aspect of an application with respect to a user, an application may be written in a manner which is independent of the content/application logic and presentation. The "interaction-based programming components" recited in independent claims 1, 44, 90 and 91 provide such advantages since, as expressly recited, they are independent of content/application logic and presentation requirements associated with the application and they may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser. Examples of such presentation requirements and modality-specific renderings include visual-based (e.g., text and graphical) renderings, speech based renderings, and combinations thereof.

Papierniak is significantly different than the claimed invention. Papierniak discloses at column 9, lines 25-53:

> FIG. 6 depicts a flow chart for the translation of information into multiple media variations with different views for a distributed on-line, and interactive environment in accordance with

a second embodiment of the invention. In FIG. 6, the Web Synchronizer receives View A which is text in Step S30. View A can be received, for example, as text, and either stored as text and translated when requested, or translated immediately into views and then stored. In Step S32, each kind of user is assigned a user type, e.g., User 1, User 2, User 3, User 4. In Step S34, the presentation is set in the Web Synchronizer/Translator. Each user is prompted for or assigned a view, e.g., Video--User 1, Audio--User 2 Graphics--User 3, Text--User 4, or a combination of these views.

A Person logs-on and is assigned or selects a user type, e.g., User 1, in Step S36, and a user, for example, User 1 requests View A in Step S38. View A is converted to, for example, video and User 1 receives the video presentation of View A in Step S40. In Step S42, a second person that may log-on the Web server is assigned or selects a user type, e.g., User 2. User 2 requests View A in Step S44, and the Web synchronizer/translator converts or retrieves View A to Audio which is then received by User 2 as an Audio Presentation of View A in Step S46. The process then continues on with other user request in Step S48, and control is returned to Step S36. Note that the translation/conversion among multiple media variations can occur either on-line or off-line, at either the information source(s) or web synchronizer/translator.

Thus, while Papierniak describes different "views" as being assigned to different users, these "views" correspond to modality-specific data, i.e., audio, graphic, and text. Such data is provided by modality-specific information sources. That is, column 8, lines 53-55, of Papierniak explains that "[t]he components of these information sources can be text-based, audio-based, video-based, image-based, graphic-based, and, in general, multimedia-based." Thus, unlike the claimed invention, no where does Papierniak disclose that "interaction-based programming components are independent of any modality and any modality-specific browser.

Even in the embodiment of FIG. 7 of Papierniak where it appears that text data is processed by a "web presence synchronizer," the initial information from the sources is still in the modality-specific form of text. That is, while Papierniak discloses a technique for translation of information into multiple media variations, the information before the translation is still in a modality-specific (text) form, and is merely translated to another modality-specific format (audio, graphic, etc.).Thus, it cannot be concluded that the data or any programming components described in Papierniak are independent of any modality, as in the claimed invention.

For at least the above reasons, Appellants assert that independent claims 1, 44, 90 and 91 are patentable over Papierniak.

Further, Appellants assert that dependent claims 2-12, 29, 36-43, 45-56, 73 and 80-87 are patentable over Papierniak not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

By way of example only, claims 8 and 52 recite wherein the user interactions are declaratively represented by the interaction-based programming components. Claims 9 and 53 recite wherein the user interactions are imperatively represented by the interaction-based programming components. Claims 10 and 54 recite wherein the user interactions are declaratively and imperatively represented by the interaction-based programming components. Claims 11 and 55 recite wherein the interaction-based programming components comprise basic elements associated with a dialog that may occur between the user and the one or more computer-based devices. Claims 12 and 56 recite wherein the interaction-based programming components comprise complex elements, the complex elements being aggregations of two or more of the basic elements associated with the dialog that may occur between the user and the one or more computer-based devices. The Office Action repeatedly cites column 9, lines 25-53, of Papierniak (copied in its entirety and addressed above) in rejecting the claims. However, nowhere can Appellants find any such features disclosed therein.

Similarly, claims 80 through 87 are respectively rejected in the Office Action by citation to column 9, lines 25-53, of Papierniak. However, nowhere does Papierniak therein disclose the claimed features.

Accordingly, withdrawal of the §102(e) rejections is respectfully requested.

(2) Claims 5 and 49 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of "New VXML Forum," posted at Cover Pages Hosted by Oasis (hereinafter "New VXML Forum").

Regarding the §103(a) rejections of claims 5 and 49 under 35 U.S.C. §103(a) based on combination of Papierniak and the New VXML Forum, Appellants assert that said claims are

patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

Further, there is a clear lack of motivation to combine the references. The Federal Circuit has stated that when patentability turns on the question of obviousness, the obviousness determination "must be based on objective evidence of record" and that "this precedent has been reinforced in myriad decisions, and cannot be dispensed with." *In re Lee*, 277 F.3d 1338, 1343 (Fed. Cir. 2002). Moreover, the Federal Circuit has stated that "conclusory statements" by an examiner fail to adequately address the factual question of motivation, which is material to patentability and cannot be resolved "on subjective belief and unknown authority." *Id.* at 1343-1344.

In the Office Action at page 6, the Examiner provides the following statement to prove motivation to combine Papierniak and the New VXML Forum, with emphasis supplied: "[i]t would have been obvious for one of ordinary skill in the Computer Networking art at the time of the invention to combine the teachings of Papierniak regarding a speech applications system with VoiceXML because VoiceXML is a form of SGML document."

Appellants submit that this statement is based on the type of "subjective belief and unknown authority" that the Federal Circuit has indicated provides insufficient support for an obviousness rejection. More specifically, the Examiner fails to identify any objective evidence of record which supports the proposed combination.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

(3) Claims 13-28, 34, 57-72 and 78 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,269,336 to Ladd et al. (hereinafter "Ladd").

Regarding the §103(a) rejections of claims 13-28, 34, 57-72 and 78 under 35 U.S.C. §103(a) based on combination of Papierniak and Ladd, Appellants assert that said claims are patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

By way of example, claims 13-28 and 57-72 recite various detailed features of the conversational gestures that make up the language of the invention. The Office Action cites code in Ladd at column 12, lines 30-67, in rejecting the majority of the claims. The other claims are rejected based on column 18, lines 56-65. However, column 12, lines 30-67, discloses "syntax and grammar that the parser unit of the voice browser utilizes to build a tree structure of the markup language document." None of this code, nor anything at column 18, lines 56-65, discloses the features of the claimed conversational gestures of the invention.

Further, based on the rationale given by the Examiner at page 7, there is a clear lack of motivation to combine the references based on the rationale of *In re Lee*, as cited above.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

(4) Claims 31, 32, 75 and 76 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,569,207 to Sundarsesan (hereinafter "Sundarsesan").

Regarding the §103(a) rejections of claims 31, 32, 75 and 76 under 35 U.S.C. §103(a) based on combination of Papierniak and Sundarsesan, Appellants assert that said claims are patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

Further, based on the rationale given by the Examiner at page 9, there is a clear lack of motivation to combine the references based on the rationale of *In re Lee*, as cited above.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

(5) Claims 30 and 74 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of World Wide Web Consortium document entitled "Extensible Stylesheet Language (XSL) version 1.0" (hereinafter referred to as "W3C XSL specification").

Regarding the §103(a) rejections of claims 30 and 74 under 35 U.S.C. §103(a) based on combination of Papierniak and W3C XSL specification, Appellants assert that said claims are patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

Further, based on the rationale given by the Examiner at page 10, there is a clear lack of motivation to combine the references based on the rationale of *In re Lee*, as cited above.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

(6) Claims 33, 77 and 88 are rejected under 35 U.S.C. §103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,493,758 to McLain (hereinafter "McLain").

Regarding the §103(a) rejections of claims 33, 77 and 88 under 35 U.S.C. §103(a) based on combination of Papierniak and McLain, Appellants assert that said claims are patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

Claims 33 and 77 recite wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices. Claim 88 recites wherein the one or more processors are distributed over the one or more computer-based devices and the application is synchronized across the one or more computer-based devices.

The only synchronization that McLain discloses (column 3, line 50, through column 4, line 6) is that of synchronizing information between a desktop computer 16 and a mobile device 18. This is not unlike "hot-syncing" a personal digital assistant (PDA) with a desktop so that the PDA has the up-to-date data from the desktop. This is not the same as representation by the interaction-based programming components permitting synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices, as recited in claims 33 and 77, or the one or more processors being distributed over the one or more computer-based devices and the application being synchronized across the one or more computer-based devices, as recited in claim 88.

Further, based on the rationale given by the Examiner at page 11, there is a clear lack of motivation to combine the references based on the rationale of *In re Lee*, as cited above.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

(7) Claims 35, 79, and 89 are rejected under 35 U.S.C. 103(a) as being unpatentable over Papierniak in view of U.S. Patent No. 6,456,974 to Baker et al. (hereinafter "Baker").

Regarding the §103(a) rejections of claims 35, 79, and 89 under 35 U.S.C. §103(a) based on combination of Papierniak and Baker, Appellants assert that said claims are patentable over the combination not only because they respectively depend from independent claims 1 and 44, but also because said claims recite patentable subject matter in their own right.

Claims 35 and 79 recite the step of including code in the authored application for permitting cosmetic altering of a presentational feature associated with the one or more modality-specific renderings of the application on the one or more computer-based devices. Claim 89 recites wherein the representation of the application further permits cosmetization of the one or more modality-specific renderings via one or more modality-specific markup languages.

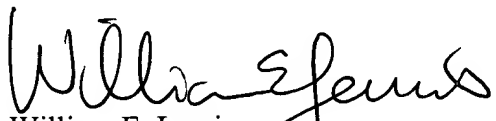
The Office Action cites column 3, lines 7-32, of Baker. However, nowhere therein does Baker mention the claimed cosmetization features. The only things mentioned in Baker therein are "scroll-up" and "go back" commands.

Further, based on the rationale given by the Examiner at page 11, there is a clear lack of motivation to combine the references based on the rationale of *In re Lee*, as cited above.

Accordingly, withdrawal of the §103(a) rejections is respectfully requested.

In view of the above, Appellants believe that claims 1-91 are in condition for allowance, and respectfully request favorable reconsideration.

Respectfully submitted,



Date: May 30, 2006

William E. Lewis
Attorney for Appellant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

CLAIMS APPENDIX

1. A method of generating an application accessible by a user through one or more computer-based devices, the method comprising the steps of:

representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that the interaction-based programming components are independent of any modality and any modality-specific browser; and

authoring the application using at least a portion of the interaction-based programming components.

2. The method of claim 1, in a client/server arrangement wherein at least a portion of the application is to be downloaded from a server to at least one of the one or more computer-based devices, acting as a client, further comprising the step of including code in the application operative to provide a connection to the content/application logic resident at the server.

3. The method of claim 2, wherein the code in the application operative to provide a connection to the content/application logic expresses at least one of one or more data models, attribute constraints and validation rules associated with the application.

4. The method of claim 1, wherein the one or more modality-specific renderings comprise a speech-based representation of portions of the application.

5. The method of claim 4, wherein the speech-based representation is based on VoiceXML.

6. The method of claim 1, wherein the one or more modality-specific renderings comprise a visual-based representation of portions of the application.

7. The method of claim 6, wherein the visual-based representation is based on at least one of HTML, CHTML and WML.

8. The method of claim 1, wherein the user interactions are declaratively represented by the interaction-based programming components.

9. The method of claim 1, wherein the user interactions are imperatively represented by the interaction-based programming components.

10. The method of claim 1, wherein the user interactions are declaratively and imperatively represented by the interaction-based programming components.

11. The method of claim 1, wherein the interaction-based programming components comprise basic elements associated with a dialog that may occur between the user and the one or more computer-based devices.

12. The method of claim 11, wherein the interaction-based programming components comprise complex elements, the complex elements being aggregations of two or more of the basic elements associated with the dialog that may occur between the user and the one or more computer-based devices.

13. The method of claim 1, wherein one of the interaction-based programming components represent conversational gestures.

14. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating informational messages to the user.

15. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating contextual help information.

16. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating actions to be taken upon successful completion of another gesture.

17. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating yes or no based questions.

18. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating dialogues where the user is expected to select from a set of choices.

19. The method of claim 18, wherein the select gesture comprises a subelement that represents the set of choices.

20. The method of claim 18, wherein the select gesture comprises a subelement that represents a test that the selection should pass.

21. The method of claim 20, wherein the select gesture comprises a subelement that represents an error message to be presented if the test fails.

22. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating rules for validating results of a given conversational gesture.

23. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating grammar processing rules.

24. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating dialogues that help the user navigate through portions of the application.

25. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for at least one of user login and authentication information.

26. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for constrained user input.

27. The method of claim 13, wherein the conversational gestures comprise a gesture for encapsulating a request for unconstrained user input.

28. The method of claim 13, wherein the conversational gestures comprise a gesture for controlling submission of information.

29. The method of claim 1, further comprising the step of defining logical input events and an association between the logical input events and physical input events that trigger the defined logical input events, such that the application may be authored using at least a portion of the definitions.

30. The method of claim 1, wherein the component by component transcoding is performed in accordance with XSL transformation rules.

31. The method of claim 1, wherein the component by component transcoding is performed in accordance with Java Bean.

32. The method of claim 1, wherein the component by component transcoding is performed in accordance with Java Server Pages.

33. The method of claim 1, wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.

34. The method of claim 1, wherein representation by the interaction-based programming components supports a natural language understanding environment.

35. The method of claim 1, further comprising the step of including code in the authored application for permitting cosmetic altering of a presentational feature associated with the one or more modality-specific renderings of the application on the one or more computer-based devices.

36. The method of claim 1, further comprising the step of including code in the authored application for permitting changes to rules for transcoding on a component by component basis to generate the one or more modality-specific renderings of the application on the one or more computer-based devices.

37. The method of claim 1, wherein a definition of an underlying data model being populated is separated from a markup language defining the user interaction.

38. The method of claim 1, wherein a node_id attribute is attached to each component and the attribute is mapped over to various outputs.

39. The method of claim 1, wherein an author is provided with a pass through mechanism to encapsulate modality-specific markup components.

40. The method of claim 1, wherein the components may be active in parallel.

41. The method of claim 1, wherein the representation and transcoding is extensible.

42. The method of claim 1, wherein a state of the application is encapsulated.

43. The method of claim 1, wherein the representation permits reference to dynamically generated data and supports callback mechanisms to the content/application logic.

44. Apparatus for use in accessing an application in association with one or more computer-based devices, the apparatus comprising:

one or more processors operative to: (i) obtain the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and (ii) transcode the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modaility-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser.

45. The apparatus of claim 44, wherein the one or more processors are distributed over the one or more computer-based devices.

46. The apparatus of claim 44, in a client/server arrangement wherein at least a portion of the application is to be downloaded from a server to at least one of the one or more computer-based

devices, acting as a client, further comprising the step of including code in the application operative to provide a connection to the content/application logic resident at the server.

47. The apparatus of claim 46, wherein the code in the application operative to provide a connection to the content/application logic expresses at least one of one or more data models, attribute constraints and validation rules associated with the application.

48. The apparatus of claim 44, wherein the one or more modality-specific renderings comprise a speech-based representation of portions of the application.

49. The apparatus of claim 48, wherein the speech-based representation is based on VoiceXML.

50. The apparatus of claim 44, wherein the one or more modality-specific renderings comprise a visual-based representation of portions of the application.

51. The apparatus of claim 50, wherein the visual-based representation is based on at least one of HTML, CHTML and WML.

52. The apparatus of claim 44, wherein the user interactions are declaratively represented by the interaction-based programming components.

53. The apparatus of claim 44, wherein the user interactions are imperatively represented by the interaction-based programming components.

54. The apparatus of claim 44, wherein the user interactions are declaratively and imperatively represented by the interaction-based programming components.

55. The apparatus of claim 44, wherein the interaction-based programming components comprise basic elements associated with a dialog that may occur between the user and the one or more computer-based devices.

56. The apparatus of claim 55, wherein the interaction-based programming components comprise complex elements, the complex elements being aggregations of two or more of the basic elements associated with the dialog that may occur between the user and the one or more computer-based devices.

57. The apparatus of claim 44, wherein one of the interaction-based programming components represent conversational gestures.

58. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating informational messages to the user.

59. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating contextual help information.

60. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating actions to be taken upon successful completion of another gesture.

61. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating yes or no based questions.

62. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating dialogues where the user is expected to select from a set of choices.

63. The apparatus of claim 62, wherein the select gesture comprises a subelement that represents the set of choices.

64. The apparatus of claim 62, wherein the select gesture comprises a subelement that represents a test that the selection should pass.

65. The apparatus of claim 64, wherein the select gesture comprises a subelement that represents an error message to be presented if the test fails.

66. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating rules for validating results of a given conversational gesture.

67. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating grammar processing rules.

68. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating dialogues that help the user navigate through portions of the application.

69. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for at least one of user login and authentication information.

70. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for constrained user input.

71. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for encapsulating a request for unconstrained user input.

72. The apparatus of claim 57, wherein the conversational gestures comprise a gesture for controlling submission of information.

73. The apparatus of claim 44, further comprising the step of defining logical input events and an association between the logical input events and physical input events that trigger the defined logical input events, such that the application may be authored using at least a portion of the definitions.

74. The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with XSL transformation rules.

75. The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with Java Bean.

76. The apparatus of claim 44, wherein the component by component transcoding is performed in accordance with Java Server Pages.

77. The apparatus of claim 44, wherein representation by the interaction-based programming components permits synchronization of the one or more modality-specific renderings of the application on the one or more computer-based devices.

78. The apparatus of claim 44, wherein representation by the interaction-based programming components supports a natural language understanding environment.

79. The apparatus of claim 44, further comprising the step of including code in the authored application for permitting cosmetic altering of a presentational feature associated with the one or more modality-specific renderings of the application on the one or more computer-based devices.

80. The apparatus of claim 44, further comprising the step of including code in the authored application for permitting changes to rules for transcoding on a component by component basis to generate the one or more modality-specific renderings of the application on the one or more computer-based devices.

81. The apparatus of claim 44, wherein a definition of an underlying data model being populated is separated from a markup language defining the user interaction.

82. The apparatus of claim 44, wherein a node_id attribute is attached to each component and the attribute is mapped over to various outputs.

83. The apparatus of claim 44, wherein an author is provided with a pass through mechanism to encapsulate modality-specific markup components.

84. The apparatus of claim 44, wherein the components may be active in parallel.

85. The apparatus of claim 44, wherein the representation and transcoding is extensible.

86. The apparatus of claim 44, wherein a state of the application is encapsulated.

87. The apparatus of claim 44, wherein the representation permits reference to dynamically generated data and supports callback mechanisms to the content/application logic.

88. The apparatus of claim 44, wherein the one or more processors are distributed over the one or more computer-based devices and the application is synchronized across the one or more computer-based devices.

89. The apparatus of claim 44, wherein the representation of the application further permits cosmetization of the one or more modality-specific renderings via one or more modality-specific markup languages.

90. A browser apparatus for use in providing access to an application by a user through one or more computer-based devices, comprising a machine readable medium containing computer executable code which when executed permits the implementation of the steps of:

obtaining the application from an application server, the application being programmatically represented by interactions that the user is permitted to have with the one or more computer-based devices by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application; and

transcoding the interaction-based programming components on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modaility-specific browsers associated with the one or more computer-based devices, the interaction-based programming components being independent of any modality and any modality-specific browser.

91. An article of manufacture for use in generating an application accessible by a user through one or more computer-based devices, comprising a machine readable medium containing computer executable code which when executed permits the implementation of the steps of:

representing interactions that the user is permitted to have with the one or more computer-based devices used to access the application by interaction-based programming components, wherein the interaction-based programming components are independent of content/application logic and presentation requirements associated with the application, and further wherein the interaction-based programming components may be transcoded on a component by component basis to generate one or more modality-specific renderings of the application renderable in accordance with one or more modality-specific browsers associated with the one or more computer-based devices, such that

the interaction-based programming components are independent of any modality and any modality-specific browser; and

authoring the application using at least a portion of the interaction-based programming components.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None